ロジスティック回帰

16/01/12

参考資料

ITエンジニアのための機械学習理論入門

神武里奈

ロジスティック回帰とは

ロジスティック回帰とは

パーセプトロンと同じ分類アルゴリズム

＋

最尤推定法で分割線のパラメーターを決定する

「新しく与えられたデータは$t = 1$である」

↓

「新しく与えられたデータが$t = 1$である確率は70%」

という、**確率的な推定**ができるようになる

分類問題への最尤推定法の適用

以下の3STEPで分類問題を解く

① 得られたデータが、ある属性値を持つ確率（事後確率）を設定しておき

② そこから逆に、トレーニングセットのデータが得られる確率（尤度関数）を計算する

③ そして、尤度関数が最大になるように、①に設定した確率の式に含まれるパラメーターを決定する

分類問題への最尤推定法の適用

復習
尤度関数・最尤推定法

トレーニングセットは最も発生確率が高いに違いない！

という仮説が正しいものとして、
トレーニングセットのデータが得られる確率
「尤度関数」が
最大になるようにパラメーターを決定する手法を
「最尤推定法」と呼ぶ

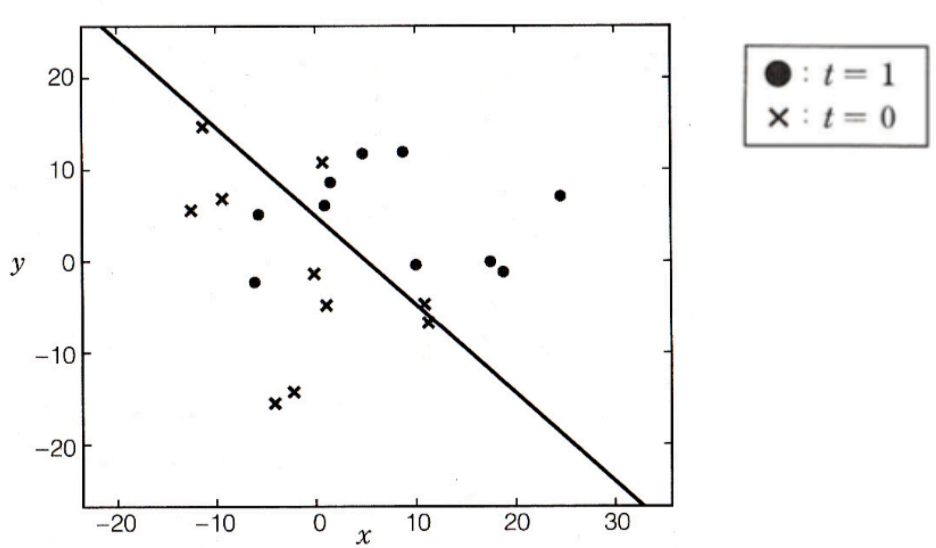
ロジスティック回帰、実例

例題

(x, y)平面上にある、
$t = 0{,}1$の属性値を持つトレーニングセットを元に、
新たなデータ(x, y)が与えられたときのtを
確率的に推定しなさい

↓

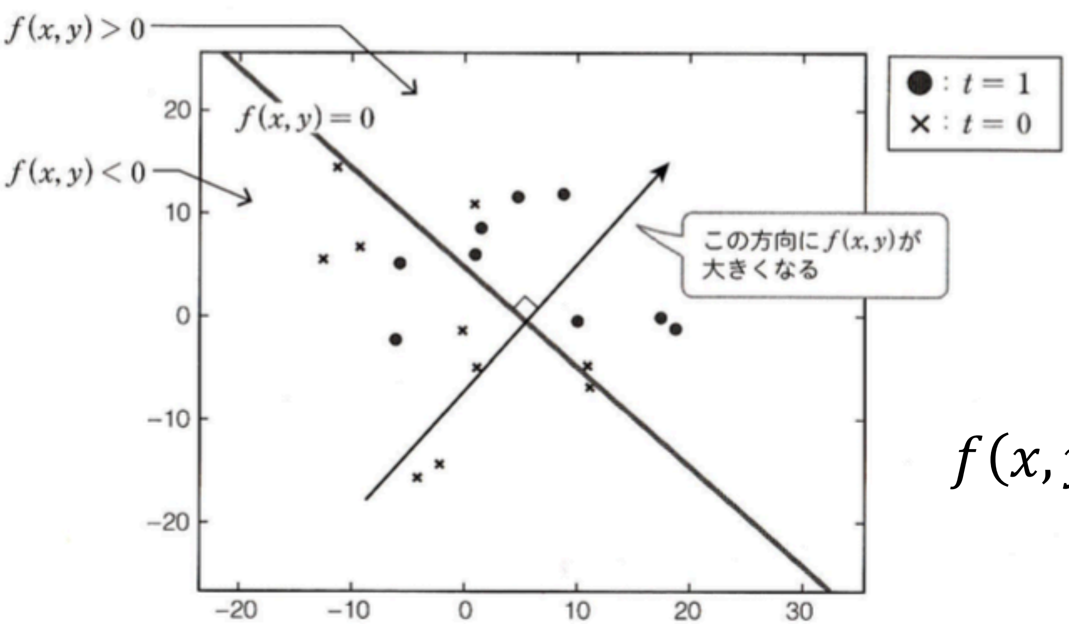
(x, y)平面上の直線を最尤推定法を用いて決定する



① 得られたデータが ある属性値を持つ確率

まずは、パーセプトロンと同様に
(x, y)平面上の直線を表す線形関数を次式で定義する

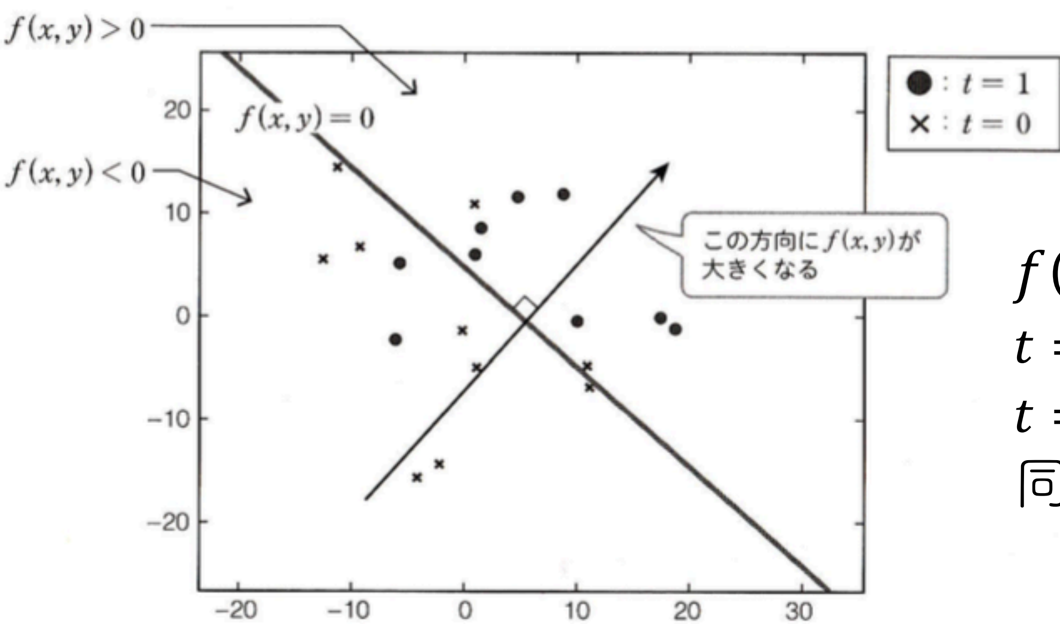
$$f(x, y) = w_0 + w_1 x + w_2 y$$



$f(x, y) = 0$ を分割線とする

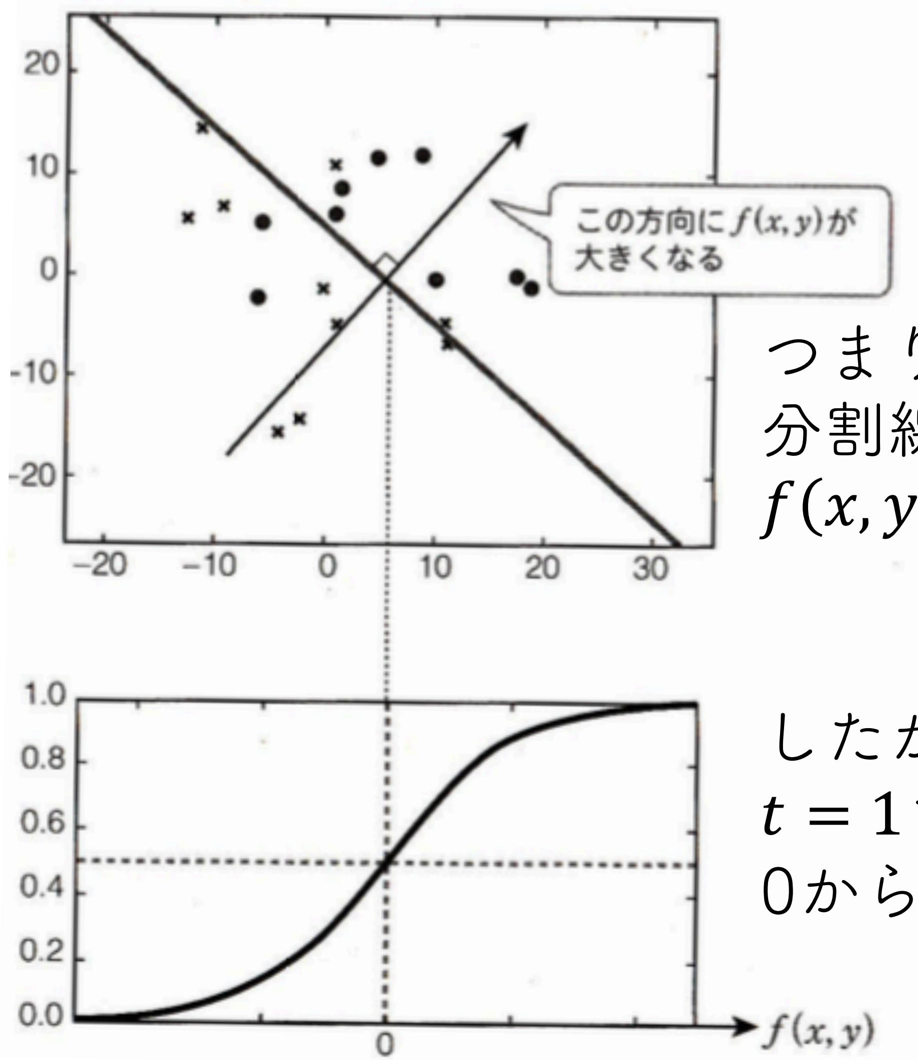
① 得られたデータが ある属性値を持つ確率

次に、(x, y)平面上の任意の点において、
得られたデータの属性が$t = 1$である確率を考える。
下図において分割線から右上の方向に離れるほど
$t = 1$である確率は高くなると考えられる。逆もしかり



$f(x, y) = 0$ 上では
$t = 1$である確率と
$t = 0$である確率は
同じ、1/2

① 得られたデータがある属性値を持つ確率



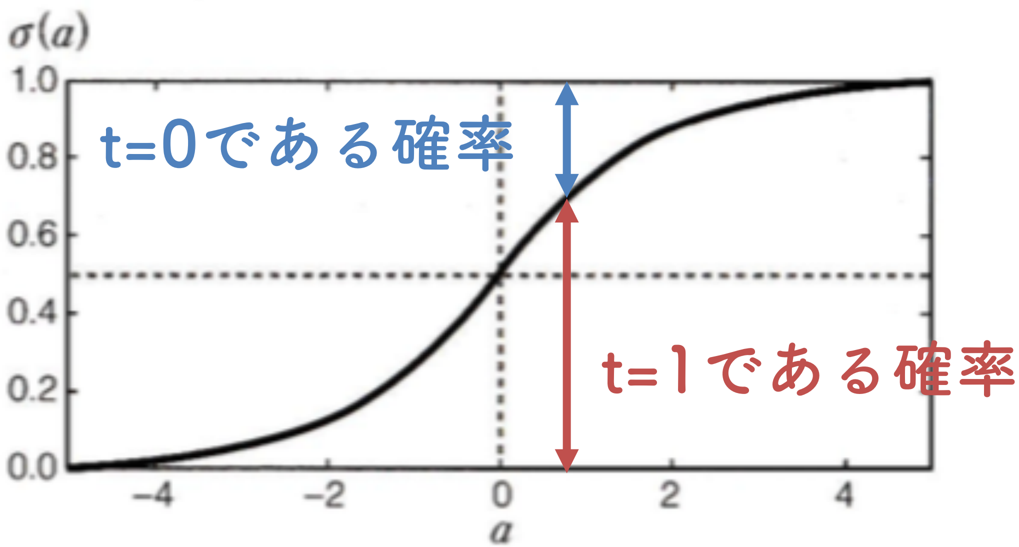
つまり
分割線からどの程度離れているかは
$f(x,y)$の値から判断することができる

したがって
$t=1$である確率は$f(x,y)$の値に従って
0から1になめらかに変化する

① 得られたデータが ある属性値を持つ確率

このように、0から1になめらかに変化するグラフは
次式のロジスティック関数で表現される

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$



① 得られたデータが ある属性値を持つ確率

以上の考察をまとめると、
(x, y)平面上の任意の点において、
得られたデータの属性が$\boldsymbol{t=1}$である確率は
次式で表される

$$P(x, y) = \sigma(w_0 + w_1 x + w_2 y)$$

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

反対に$t = 0$である確率は $1 - P(x, y)$ になる

② トレーニングセットのデータが得られる確率

この確率を元に、トレーニングセットとして与えられたデータ$\{(x_n, y_n, t_n)\}_{n=1}^{N}$が得られる確率を考える

これは既に得られた結果に対して、
後付けで確率を考えているようなもの

例：実際に2個のサイコロを振ってゾロ目が出た時に
自分はどれくらい珍しい体験をしたかを
考えるようなもの

② トレーニングセットのデータが得られる確率

この確率を元に、トレーニングセットとして与えられたデータ$\{(\boldsymbol{x_n}, \boldsymbol{y_n}, \boldsymbol{t_n})\}_{\boldsymbol{n=1}}^{\boldsymbol{N}}$が得られる確率を考える

まず、特定の1つのデータ(x_n, y_n, t_n)が得られる確率は

t_n=1の場合： $\mathrm{P}(x_n, y_n)$
t_n=0の場合： $1 - \mathrm{P}(x_n, y_n)$

この2式は次式のようにまとめて書くことができる

$$P_n = \mathrm{P}(x_n, y_n)^{t_n}\{1 - \mathrm{P}(x_n, y_n)\}^{1-t_n}$$

② トレーニングセットのデータが得られる確率

ここで、
$P_n = \mathrm{P}(x_n, y_n)^{t_n}\{1 - \mathrm{P}(x_n, y_n)\}^{1-t_n}$
に、得られたデータの属性が$t = 1$である確率の式
$\mathrm{P}(x, y) = \sigma(w_0 + w_1 x + w_2 y)$
を代入すると、P_nは次式で表すことができる

$$P_n = {Z_n}^{t_n}(1 - Z_n)^{1-t_n}$$

$$Z_n = \sigma(w^T \phi_n)$$

Z_nはn番目のデータの属性が$t = 1$である確率を表す

② トレーニングセットのデータが得られる確率

$$Z_n = \sigma(w^T \phi_n)$$

Z_nはn番目のデータの属性が$t = 1$である確率を表す
wとϕ_nはパーセプトロンの計算で用いたものと同じ

$$w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$$

$f(x, y)$の係数を並べたベクトル

$$\phi_n = \begin{pmatrix} 1 \\ x_n \\ y_n \end{pmatrix}$$

トレーニングセットにおける
n番目のデータの座標に
バイアス項を付け加えたベクトル

② トレーニングセットのデータが得られる確率

最後に、トレーニングセットに含まれるデータを
まとめて考えると、
これらが得られる確率は、各データが得られる確率
$P_n = {Z_n}^{t_n}(1-Z_n)^{1-t_n}$の積になる

$$P = \prod_{n=1}^{N} P_n = \prod_{n=1}^{N} {Z_n}^{t_n}(1-Z_n)^{1-t_n}$$

トレーニングセットが得られる確率Pを
$Z_n = \sigma(w^T \phi_n)$を通して、
パラメーターwの関数として見た上式が尤度関数である

③ 尤度関数が最大になるように パラメーターを決定

次は、尤度関数が最大になるように、
パラメーターwを決定する。

しかし、**式変形だけではwを直接求めることができない**

パーセプトロンと同様に、
確率Pの値が大きくなる方向に
wを修正する手順を繰り返すアルゴリズムを用いる必要

パーセプトロン　　　　　ロジスティック回帰
「確率的勾配降下法」→「ニュートン・ラフソン法」

③ 尤度関数が最大になるように パラメーターを決定

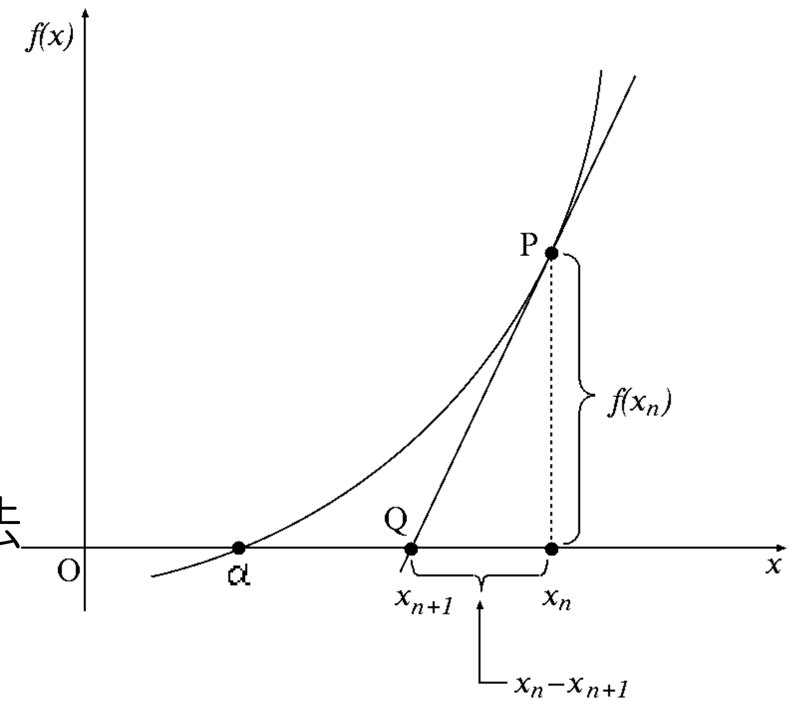
「確率的勾配降下法」（素朴）
勾配ベクトルの反対方向にパラメーターを修正する手法

「ニュートン・ラフソン法」

$f(x)=0$となるxを次式を用いて
求める「ニュートン法」

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

を多次元・非線形に拡張した手法



③ 尤度関数が最大になるように
パラメーターを決定

$$w_{new} = w_{old} - (\phi^T R\phi)^{-1}\phi^T(z - t)$$

$t = \begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix}$ トレーニングセットの属性値t_nを並べたベクトル

$\phi = \begin{pmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_N & y_N \end{pmatrix}$ 各データの座標を表すベクトルϕ_nを横ベクトルにしてならべたNx3行列

定数

$z = \begin{pmatrix} z_1 \\ \vdots \\ z_N \end{pmatrix}$ $Z_n = \sigma(w^T\phi_n)$を並べたベクトル

$R = diag[z_1(1 - z_1), \cdots, z_N(1 - z_N)]$ $z_n(1 - z_n)$を成分とする対角行列

wを**変数**に持つ

③ 尤度関数が最大になるように パラメーターを決定

$$w_{new} = w_{old} - (\phi^T R \phi)^{-1} \phi^T (z - t)$$

つまり、パラメーターw_{old}が与えられた際に、
zとRを計算しておき、
修正された新しいパラメーターw_{new}を決定する

w_{new}をw_{old}としてさらに新しいw_{new}を計算することを
繰り返すと確率Pの値が大きくなり、
最終的に最大値に達することを証明できる

*具体的な導出については
5.3 IRLS法（反復再重み付け最小二乗法）の導出を参照

③ 尤度関数が最大になるように
パラメーターを決定

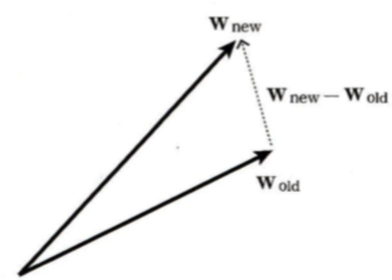
$$w_{new} = w_{old} - (\phi^T R\phi)^{-1}\phi^T(z - t)$$

また、上式の計算を繰り返すと、
Pの値が最大値に近づくにつれて、
パラメーターwの変化の割合は小さくなっていく

変化の割合が閾値を切った時点で計算を打ち切る
（一種のオーバーフィッティングを避けるため）

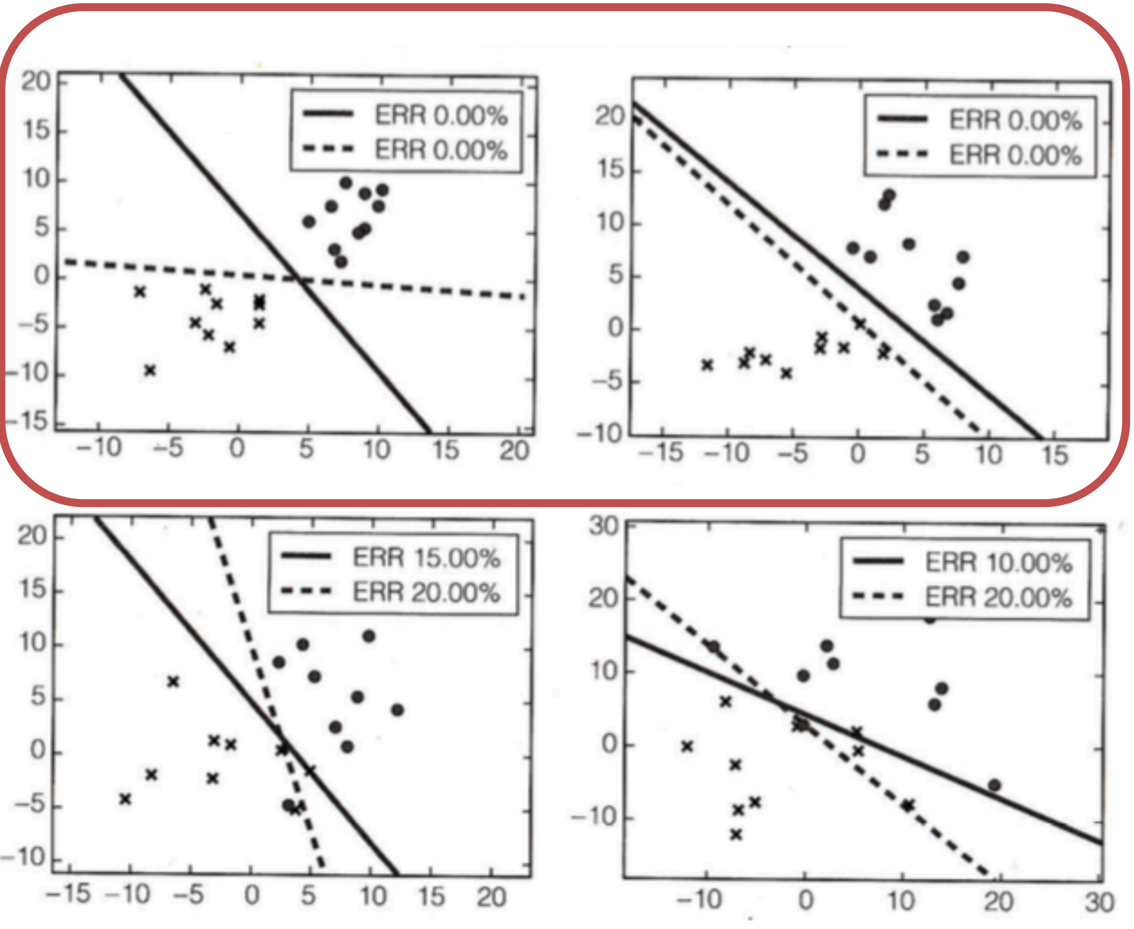
$$\frac{||w_{new} - w_{old}||^2}{||w_{old}||^2} < 0.001\ etc\ \ldots$$

これは、wをベクトルとみなした際の変化を考え
変化分のベクトルの大きさの2乗が
修正前のベクトルの大きさの0.1%未満になるという条件



ロジスティック回帰と
パーセプトロンの比較

ロジスティック回帰と
パーセプトロンの比較



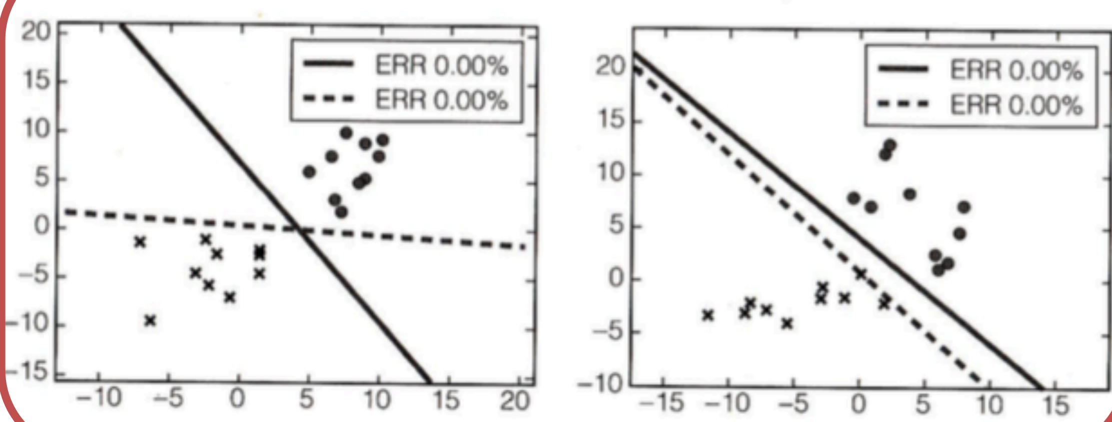
実線：ロジスティック回帰
破線：パーセプトロン

ERR：正しく分類できなかったデータの割合

条件
ロジスティック回帰では
変化の割合が閾値を切った時点で計算を終了
30回繰り返しても
閾値を切らない場合は
その時点で計算を打ち切る

パーセプトロンでは
正しく分類されていなければ
パラメーターを修正するという処理を
30回繰り返した時点で計算を終了

ロジスティック回帰と
パーセプトロンの比較



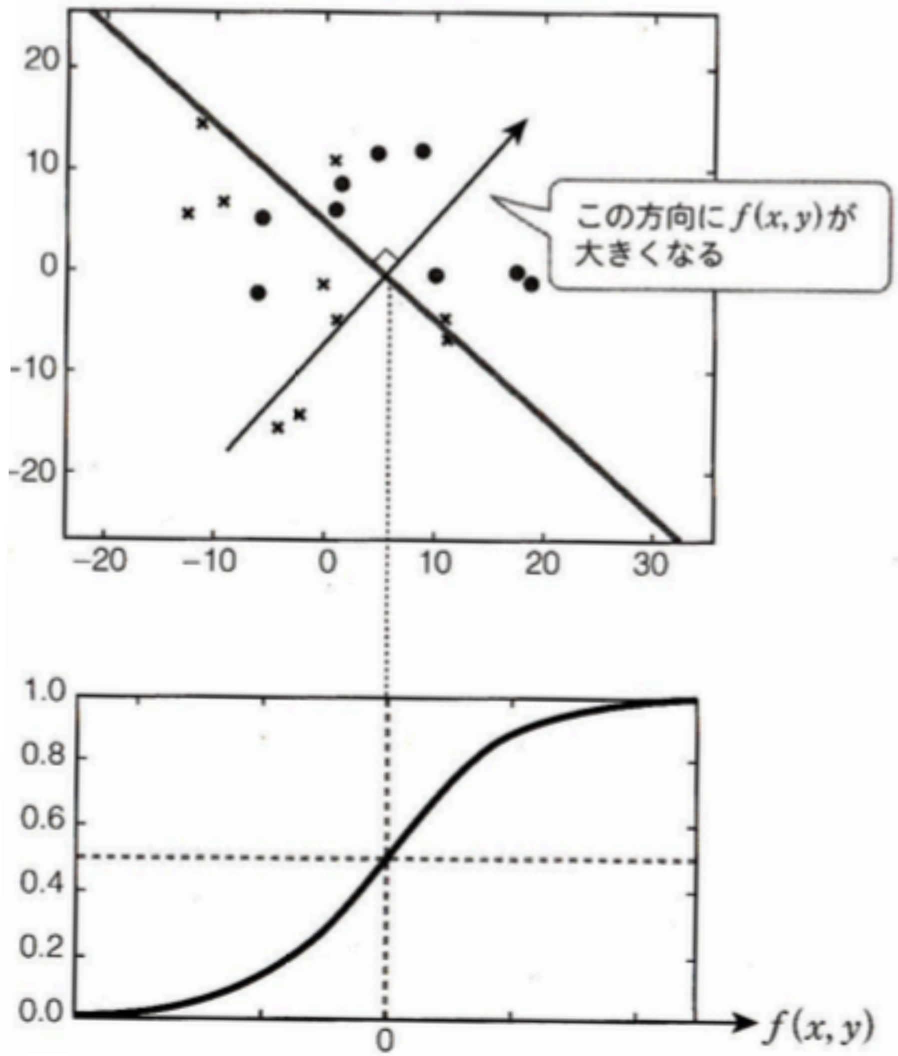
どちらも全てのデータを
正しく分類しているが
ロジスティック回帰では
それぞれの属性のデータ群の
ほぼ中央部分に分割線がある
パーセプトロンでは
少し偏った位置にある

↓

確率的勾配降下法では、一度すべてのデータが正しく分類されると、
パラメーターの変化が停止する
ロジスティック回帰では、トレーニングセットのデータが得られる
全体的な確率を最大化しようとするため、
正しく分類する中でよりもっともらしいものが選択される

ロジスティック回帰の
現実問題への適用

ロジスティック回帰の現実問題への適用



ロジスティック回帰では
得られるデータがt=1である確率
を考えることによって
分割線を決定した

$f(x,y)=0$で与えられる分割線は
確率が1/2になる点に対応する

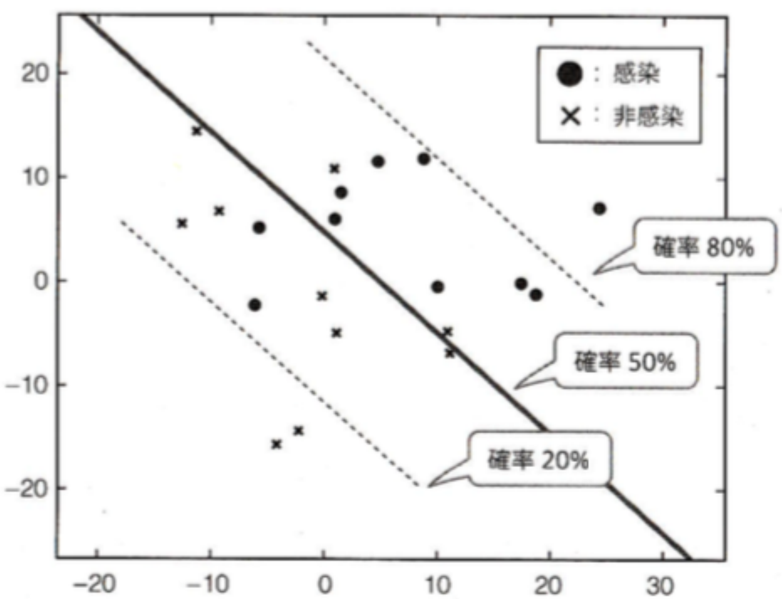
↓

ロジスティック回帰で
得られた結果を現実の問題に
適用する場合、
確率1/2を境界線にすることは
適切ではなくなくなくない？

ロジスティック回帰の現実問題への適用

現実の問題例

トレーニングセットのデータについて
(x_n, y_n)はウイルス感染の一次検査の数値で、
t_nは実際に感染していたかどうかを表す



新たな検査結果が得られた際に、
その人がウイルスに感染している確率が
20%・50%・80%と推定される直線

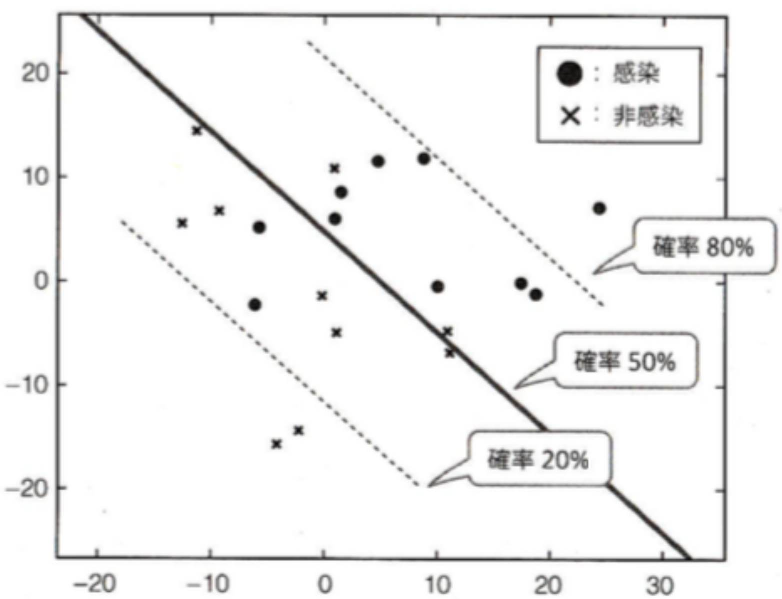
ロジスティック回帰の現実問題への適用

現実の問題例

感染確率が50%以上と推定される人には精密検査を勧告する

↓

これは本当に正しい判断か？



このような場合
適切な判断基準を見出すには
「**真陽性率**」と「**偽陽性率**」
について考える必要がある

ロジスティック回帰の
現実問題への適用

言葉の定義
一般の分類問題において、
発見したい属性を持つデータを「陽性（Positive）」
そうでないデータを「陰性（Negative）」と呼ぶ

先ほどの例では
t=1の属性を持つデータ、
すなわちウイルスに感染した人を発見することが目的
つまりt=1のデータが「陽性」

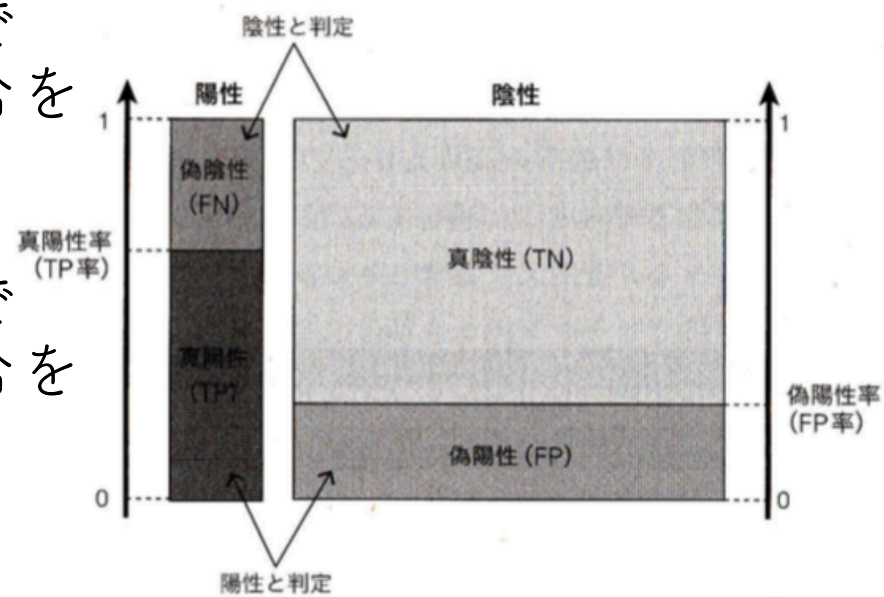
ロジスティック回帰の
現実問題への適用

新たなデータが陽性であるかどうかを判定するとき
「陽性」だと判断したデータについて、

それが本当に陽性だったものを真陽性（True Positive）
本当は陰性だったものを偽陽性（False Positive）と呼ぶ

「陽性」であったデータの中で
「真陽性」となるデータの割合を
「真陽性率（TP率）」と呼ぶ

「陰性」であったデータの中で
「偽陽性」となるデータの割合を
「偽陽性率（FP率）」と呼ぶ

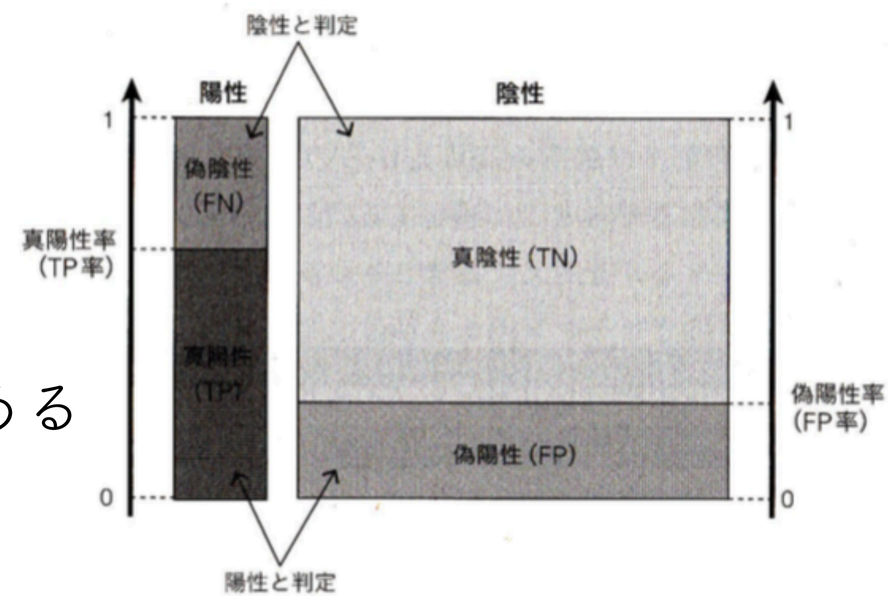


ロジスティック回帰の現実問題への適用

「真陽性率（TP率）」はウイルス感染している人の中でその何割を正しく判断することができたかを示す

「偽陽性率（FP率）」はウイルスに感染していない人中その何割を誤って感染していると判断したかを示す

医師は
真陽性率を上げ
偽陽性率を下げたい
このトレードオフを考え
判定ラインを設定する必要がある



ROC曲線による学習モデルの評価

ROC曲線は
どのような確率を境界にするのがよいかを判断する、
つまり、**真陽性率**と**偽陽性率**の関係を分析する道具

機械学習に使用したアルゴリズム（学習モデル）
そのものの良し悪しを判断することにも利用できる

ROC曲線による性能評価

① ロジスティック回帰を適用して
$f(x, y)$のパラメーター$(w_0\ , w_1\ , w_2)$を具体的に決定する

② これを$\mathrm{P}(x, y)=\ \sigma(w_0 + w_1\, x + w_2\, y)$に代入すると
座標(x, y)のデータが属性t=1を持つ確率$\mathrm{P}(x, y)$が決まる

③ トレーニングセットのそれぞれのデータについて
確率$\mathrm{P}(x_n, y_n)$を計算した後に、
確率の大きい順に並び替える

ROC曲線による性能評価

No.	x	y	t	P
1	24.43	6.95	1	0.98
2	8.84	11.92	1	0.91
3	18.69	-1.17	1	0.86
4	17.37	-0.07	1	0.86
5	4.77	11.66	1	0.85
6	0.83	10.74	0	0.73
7	1.57	8.51	1	0.69
8	10.07	-0.53	1	0.66
9	0.99	6.04	1	0.58
10	10.73	-4.88	0	0.53
11	11.16	-6.77	0	0.47
12	-11.21	14.64	0	0.46
13	-5.67	5.05	1	0.31
14	-0.06	-1.47	0	0.28
15	-9.25	6.74	0	0.26
16	1.05	-4.86	0	0.21
17	-12.35	5.61	0	0.16
18	-6.12	-2.41	1	0.12
19	-2.17	-14.40	0	0.04
20	-4.06	-15.70	0	0.02

P > 1
真陽性率 = 0
偽陽性率 = 0

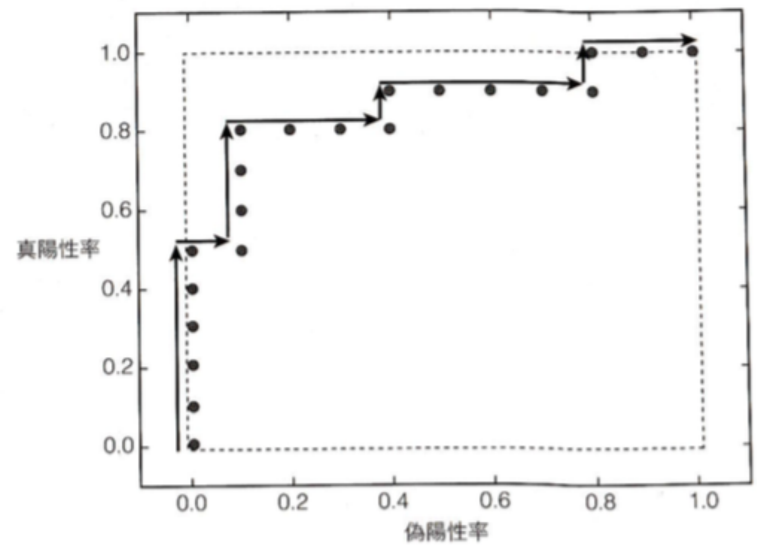
P > 0.95
真陽性率 = 1/10
偽陽性率 = 0

P > 0.90
真陽性率 = 2/10
偽陽性率 = 0

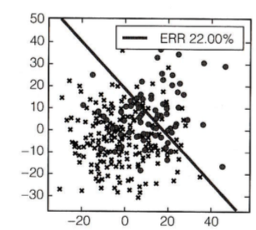
判定ラインを
1段づつ下げながら
真陽性率と偽陽性率を
計算していく

ROC曲線による性能評価

先ほどのデータの
真陽性率と偽陽性率の組のグラフ→

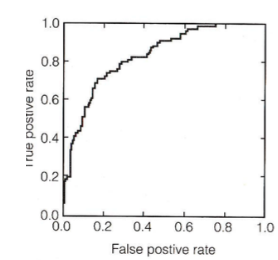


この後は実際の問題に応じて、
どの点を判断基準として選択するかを考察する



例：許容可能な偽陽性率の範囲内で、
真陽性率のなるべく高い点を選択して、
その点の確率Pを判定基準にする

たくさんのデータになると曲線のようになる→



機械学習で得られる結果と
現実のビジネスに役立つ判断指標は全くの別物

機械学習で得られた結果の意味を理解しなければ
現実の問題に適用して有益な結果を得るのは
難しいことがこの例から感じることができるのではでは